TLCharts

Doron Drusinsky

ddrusins@nps.edu

www.time-rover.com

TLCharts

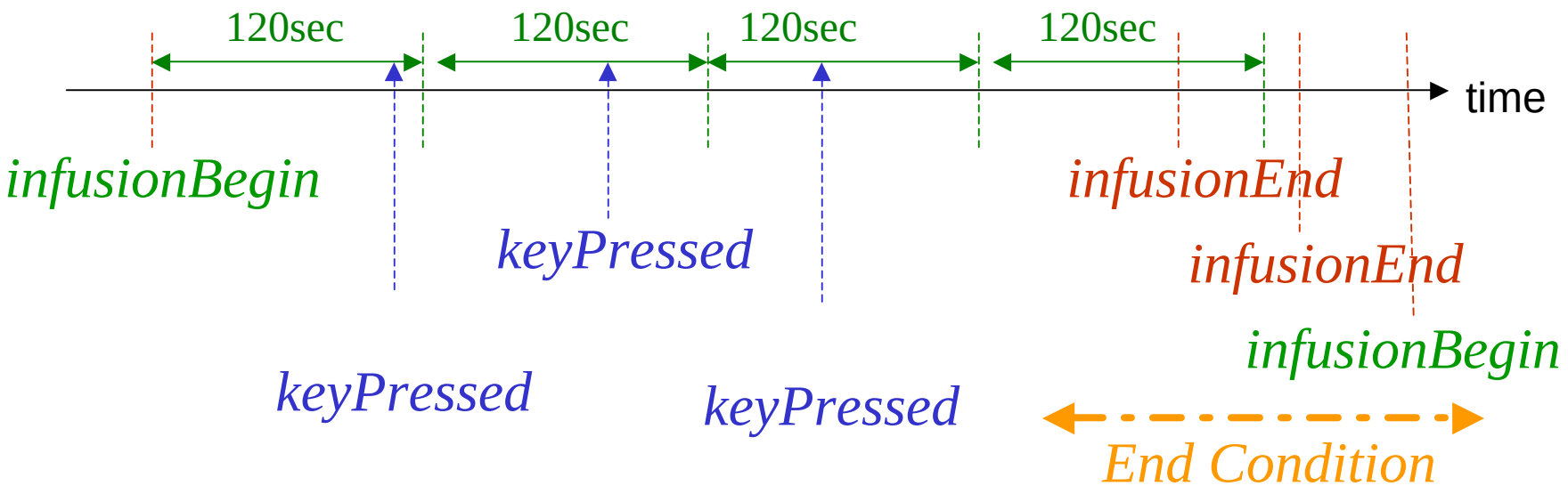
Positioning:

Customers interest

1. Spec of system (before impl./design)
2. Spec of properties in impl./design

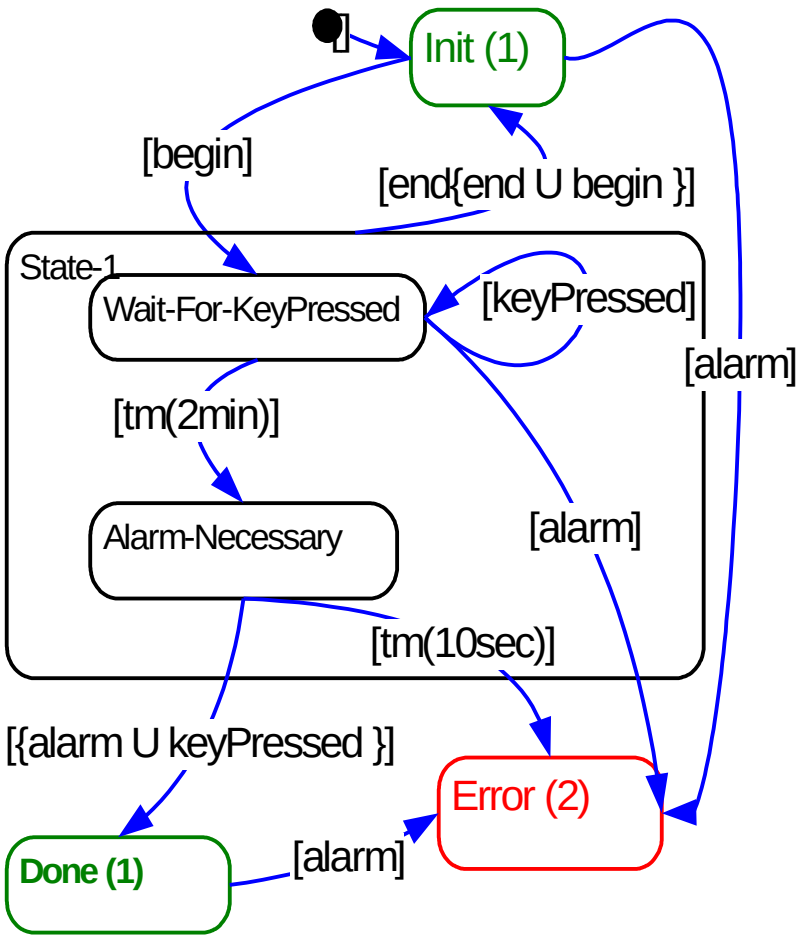
Example: Infusion Pump

between every *infusionBegin* and an *End-condition* session, a *keyPressed* must be repeatedly sensed within two-minute intervals or else an alarm must sound within 10 seconds and until *keyPressed* is sensed. Also according to this specification, once the alarm sounds then the assertion has succeeded and no more alarms are permitted. The *End-condition* is defined *infusionEnd* being sensed until *infusionBegin* is sensed.



Specification Languages: TLCharts

Harel Statecharts: Visuality, Familiarity, Readability



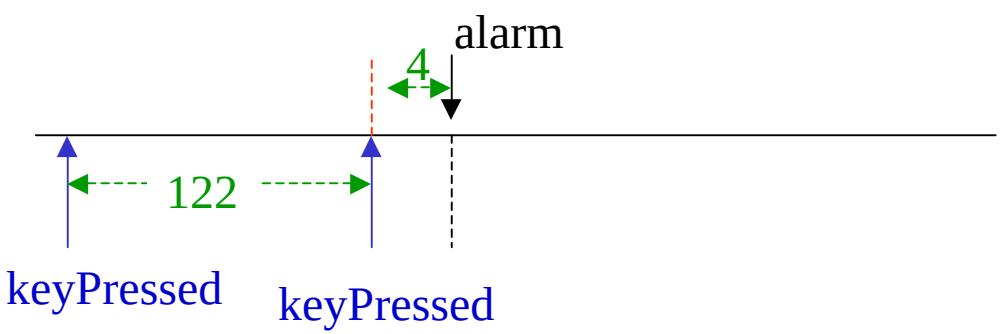
VS.

$$
\begin{aligned}
&\ddot{y}\ (\ \text{infusionBegin} \Rightarrow \\
&\quad (\ ((\text{infusionBegin} \vee \text{keyPressed}) \Rightarrow \\
&\quad\quad (\ (\ddot{y}\ \neg\text{alarm}) \wedge \\
&\quad\quad\quad ((?\ \ ?_{\leq 120}\ \text{keyPressed}) \\
&\quad\quad\quad\quad \vee (\neg\text{keyPressed}\ U_{[120,130]} \\
&\quad\quad\quad\quad\quad (\text{alarm}\ U\ \ (\text{keyPressed} \wedge \ddot{y}\neg\text{alarm}))) \\
&\quad\quad\quad) \\
&\quad\quad) \\
&\quad) \ U\ (\text{infusionEnd}\ U \\
&\quad\quad\quad (\text{infusionBegin}\ \wedge \text{infusionEnd})) \\
&))
\end{aligned}
$$

LTL Discussion

The assertion fails because LTL's ρ*Until*φ requires ρ to repeatedly succeed until φ succeeds, namely *Not keyPressed* must be true until the alarm.

$$
\begin{array}{l}
\ddot{y}\ (\ \text{infusionBegin} \Rightarrow \\
\quad (\ ((\text{infusionBegin} \vee \text{keyPressed}) \Rightarrow \\
\quad\quad (\ (\ddot{y}\ \neg\text{alarm}) \wedge \\
\quad\quad\quad ((?\ \ ?_{\leq 120}\ \text{keyPressed}) \\
\quad\quad\quad\quad \vee (\neg\text{keyPressed}\ U_{[120,130]} \\
\quad\quad\quad\quad\quad (\text{alarm}\ U\ \ (\text{keyPressed} \wedge \ddot{y}\neg\text{alarm}))) \\
\quad\quad\quad) \\
\quad\quad) \\
\quad) \ U\ (\text{infusionEnd}\ U\ \ (\text{infusionBegin}\ \wedge \text{infusionEnd})) \\
\quad))
\end{array}
$$

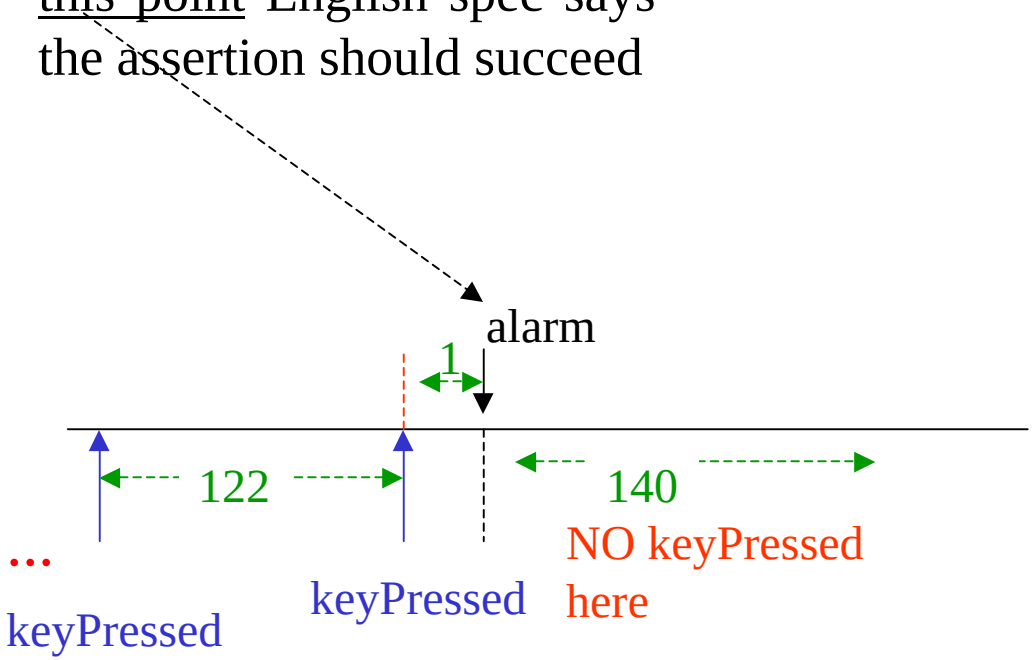


An interval of 122 seconds between two consecutive keyPressed events followed by an alarm sounding 4 seconds later

LTL Discussion

…once alarm sounds according to this specification then the assertion has succeeded.

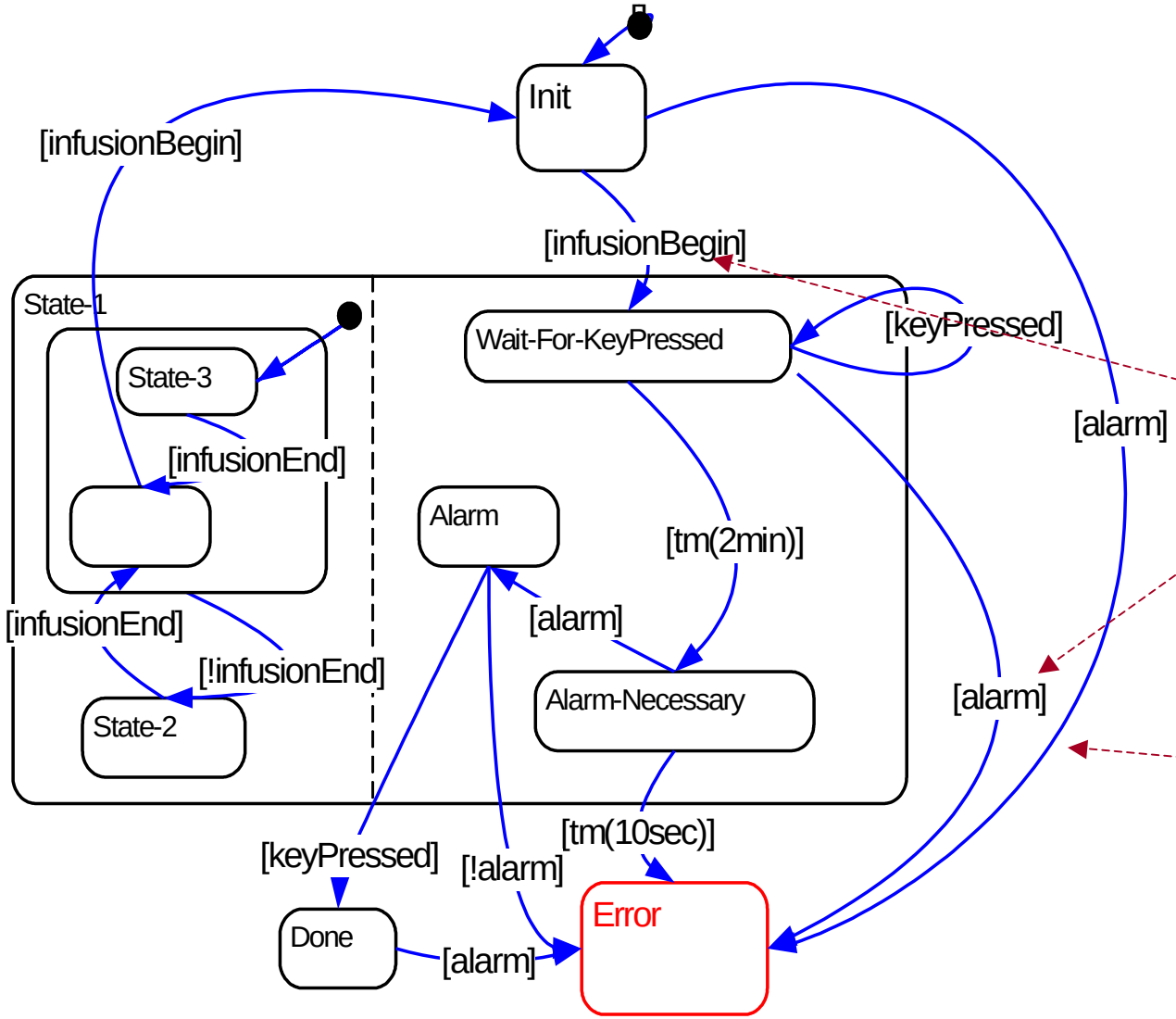
The assertion fails though at this point English spec says the assertion should succeed



*An interval of 122 seconds between two consecutive keyPressed events followed by an alarm sounding 1 second later, followed by **no** keyPressed or alarm for 140 seconds.*

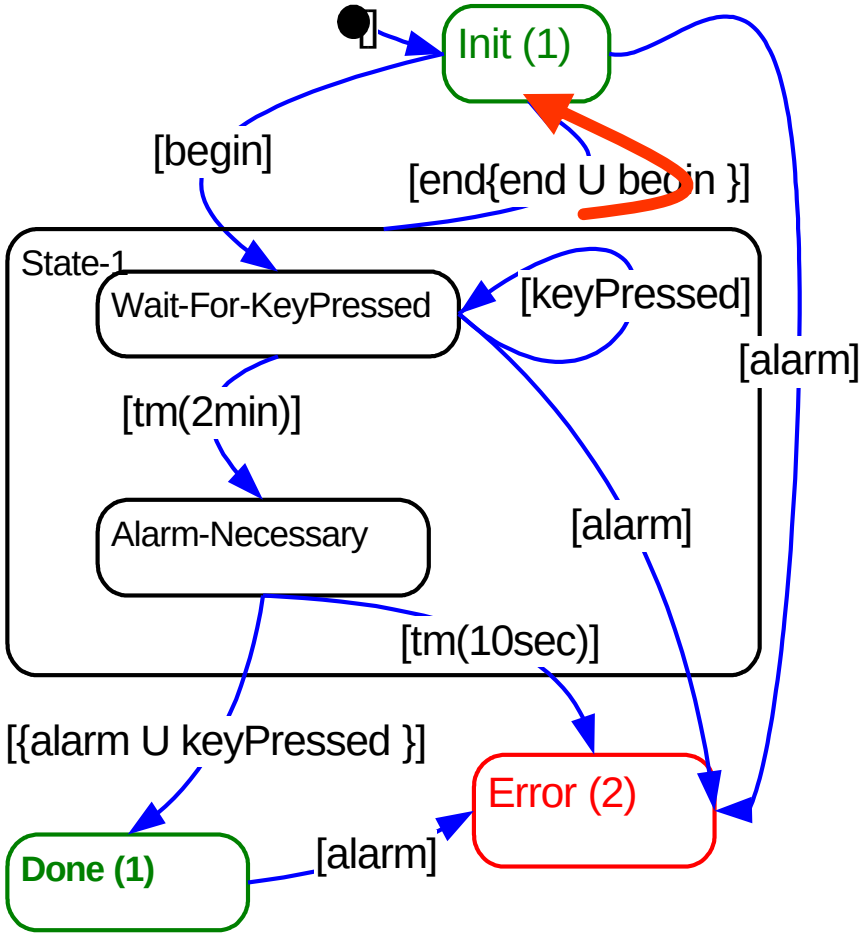
Specification Languages: statecharts

Statecharts for formal specification



- **Deterministic**
- **Asserts about sequences of inputs and corresp. sys. outputs**
- **Says Y/N (good/bad) about those combo sequences.**
- **Has negative info (what cannot happen)**

Specification Languages: TLCharts

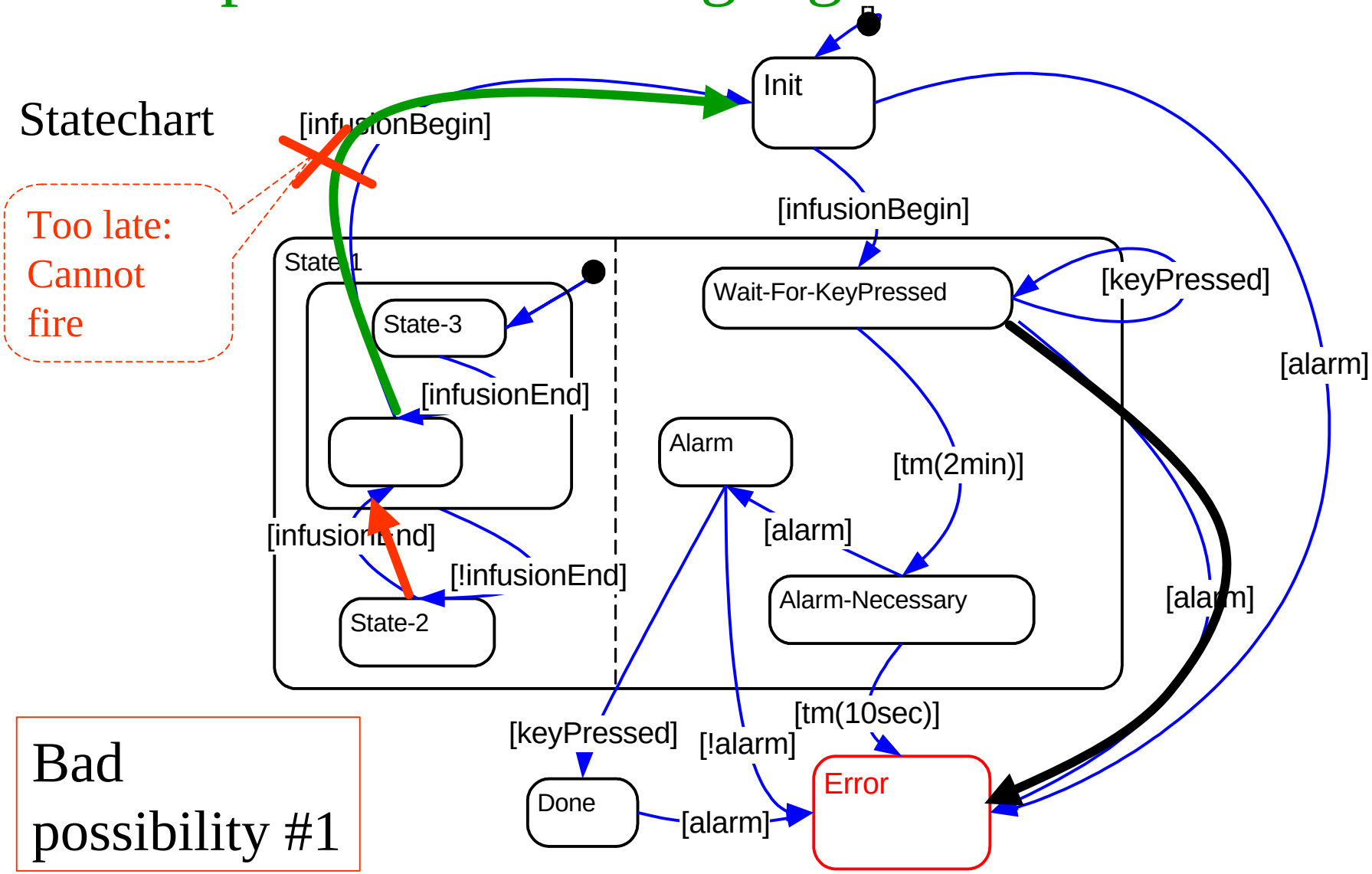


TLChart

Sequence: … *infusionEnd* . *alarm* . *keyPressed* . *infusionBegin*.

Specification Languages: TLCharts

Statechart

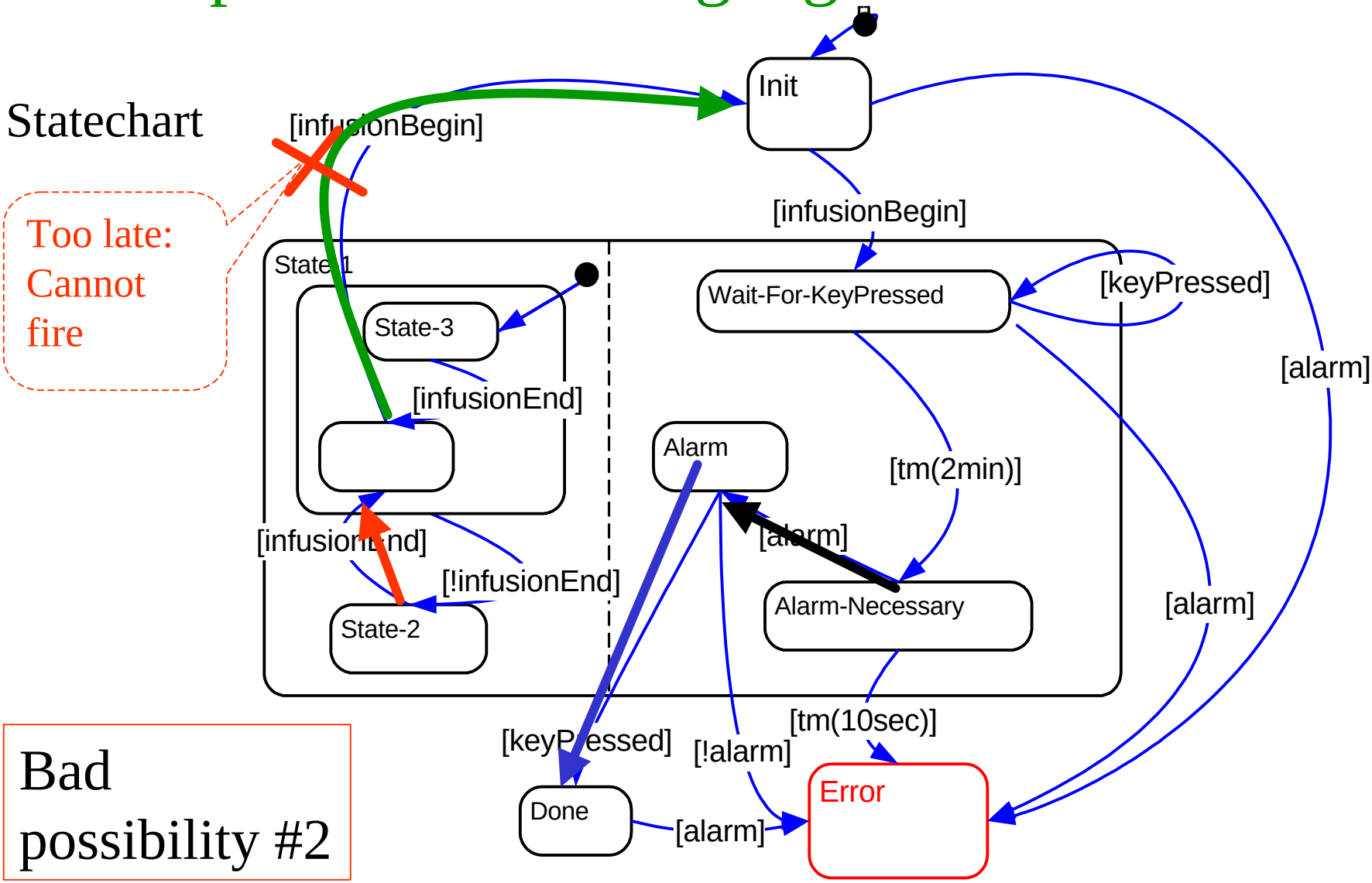


Bad possibility #1

Sequence: … *infusionEnd* . *alarm* . *keyPressed* . *infusionBegin*.

Specification Languages: TLCharts

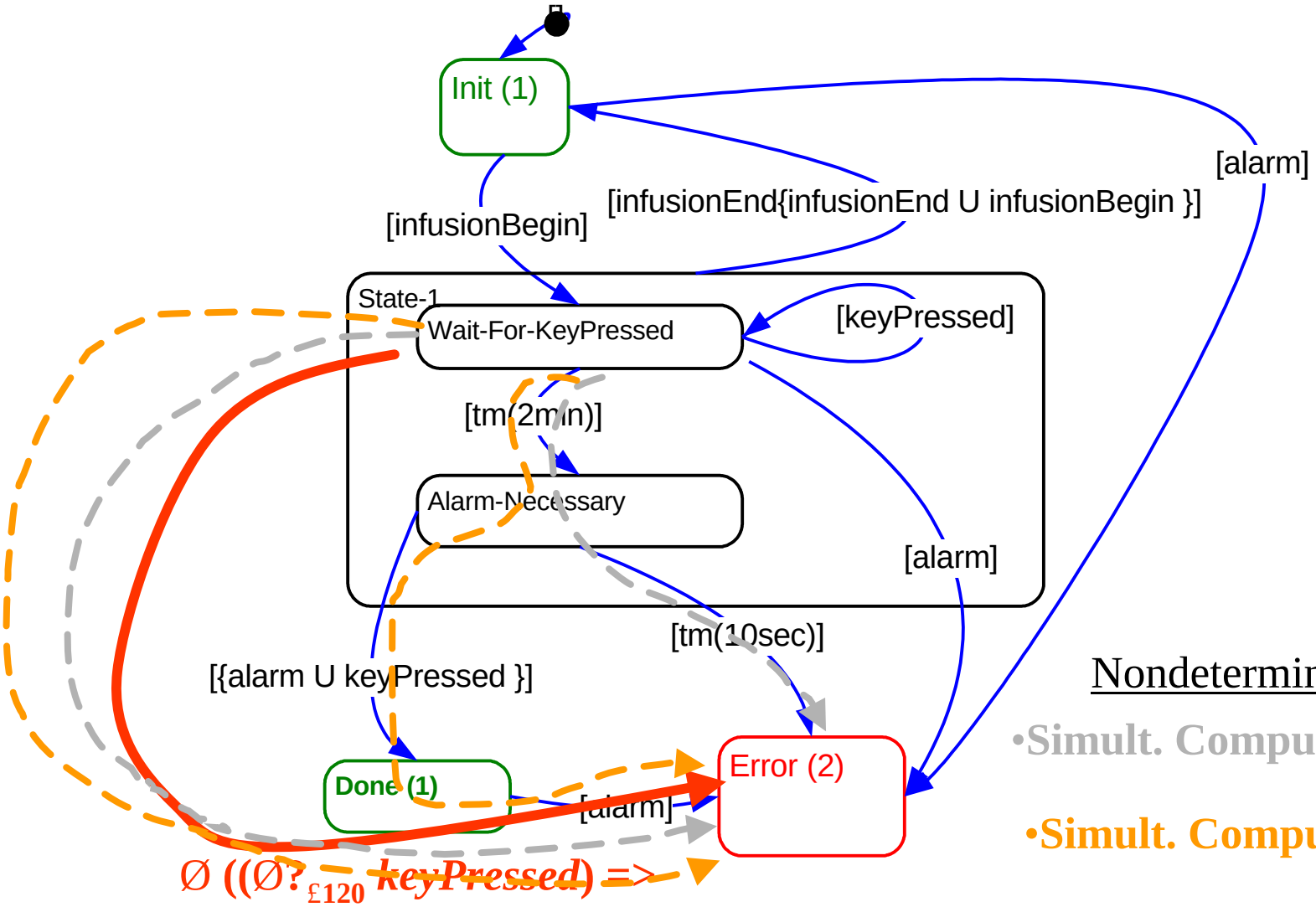
Statechart



Bad possibility #2

Sequence: … *infusionEnd* . *alarm* . *keyPressed* . *infusionBegin*.

Specification Languages: Armor plating TLCharts



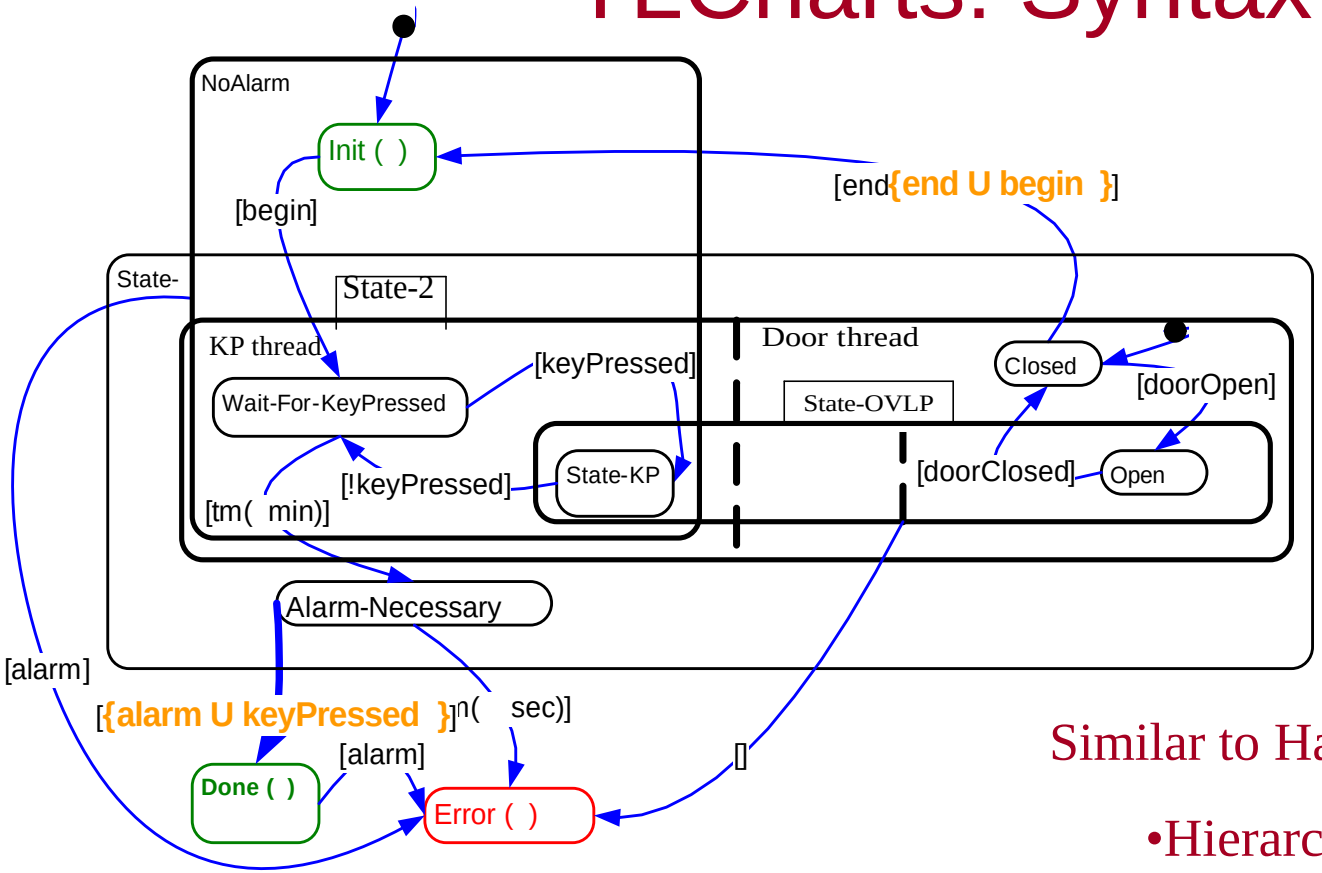
Ø ((Ø?$_{£120}$ ~~keyPressed~~) =>

?$_{[120,130]}$ (*alarm U (keyPressed* Ù ÿØ*alarm*))

Nondeterminism

- Simult. Computations #1
- Simult. Computations #2

TLCharts: Syntax



Similar to Harel Statecharts:

- Hierarchy
- Concurrence
- History states
- Events, conditions, timers.

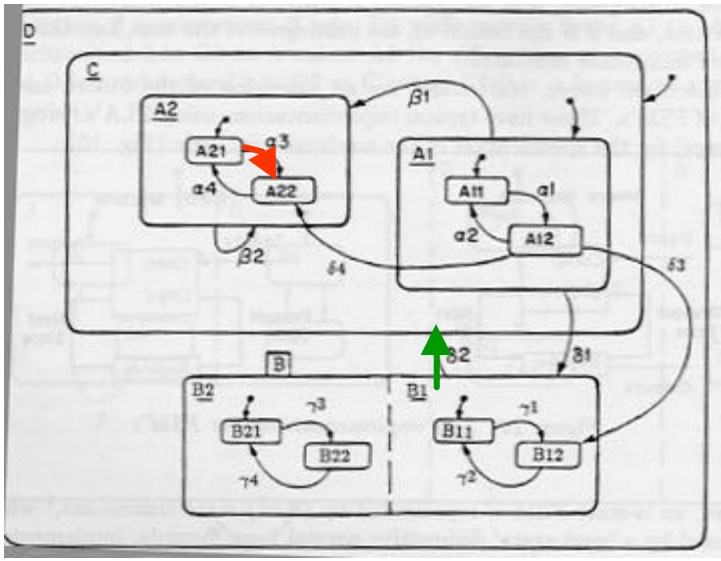
And also:

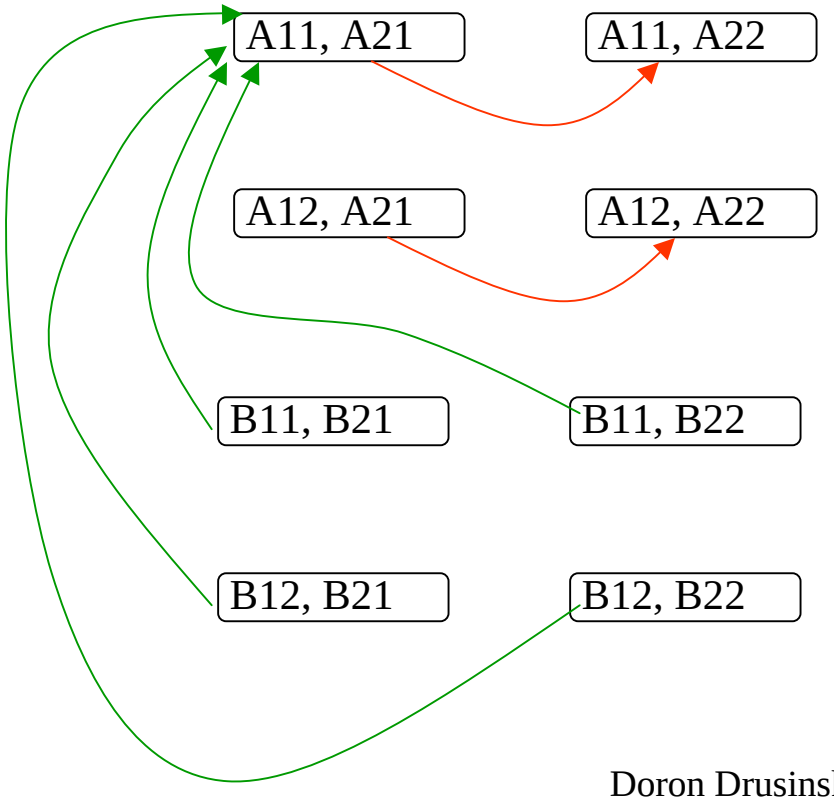
- LTL/MTL/Regex guards
- (Prioritized) good/error states
- State overlapping

TLCharts: Interlingua-based Semantics

First, consider TLCharts without LTL/regex guards:

- Defined in terms of an Equivalent NFA (**ENFA**).

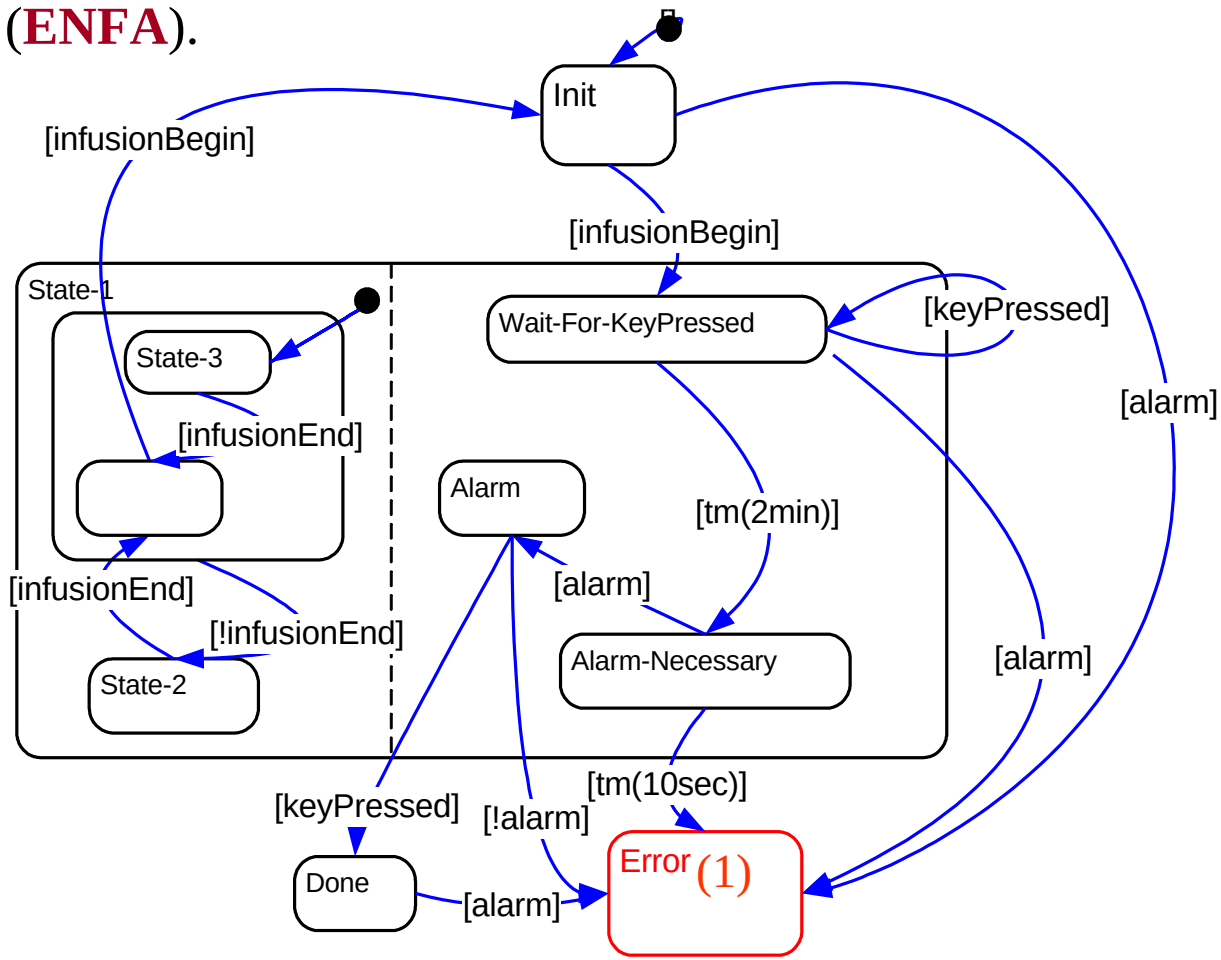




TLCharts: Interlingua-based Semantics

First, consider TLCharts without LTL/regex guards:

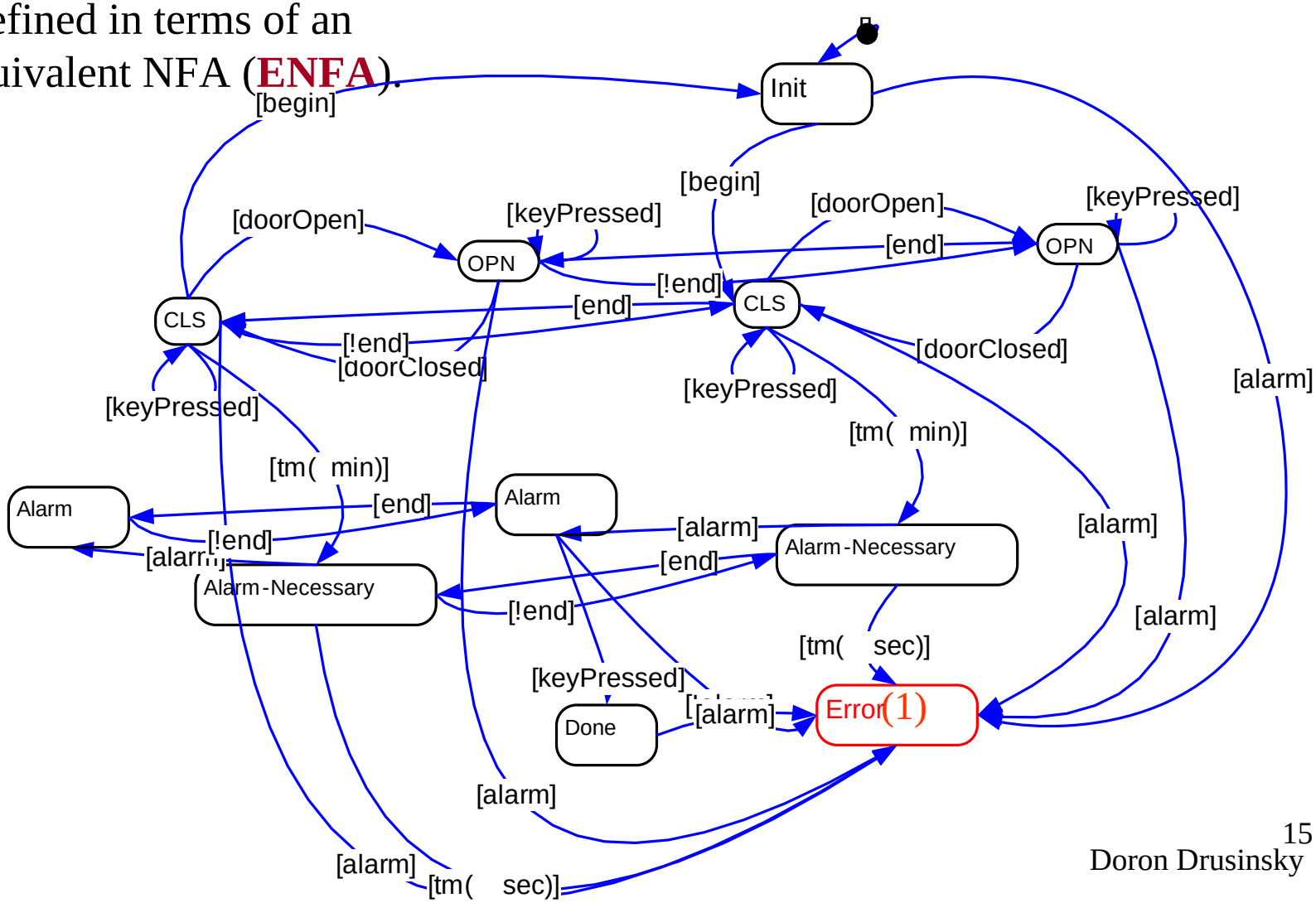
- Defined in terms of an Equivalent NFA (**ENFA**).



TLCharts: Interlingua-based Semantics

First, consider TLCharts without LTL/regex guards:

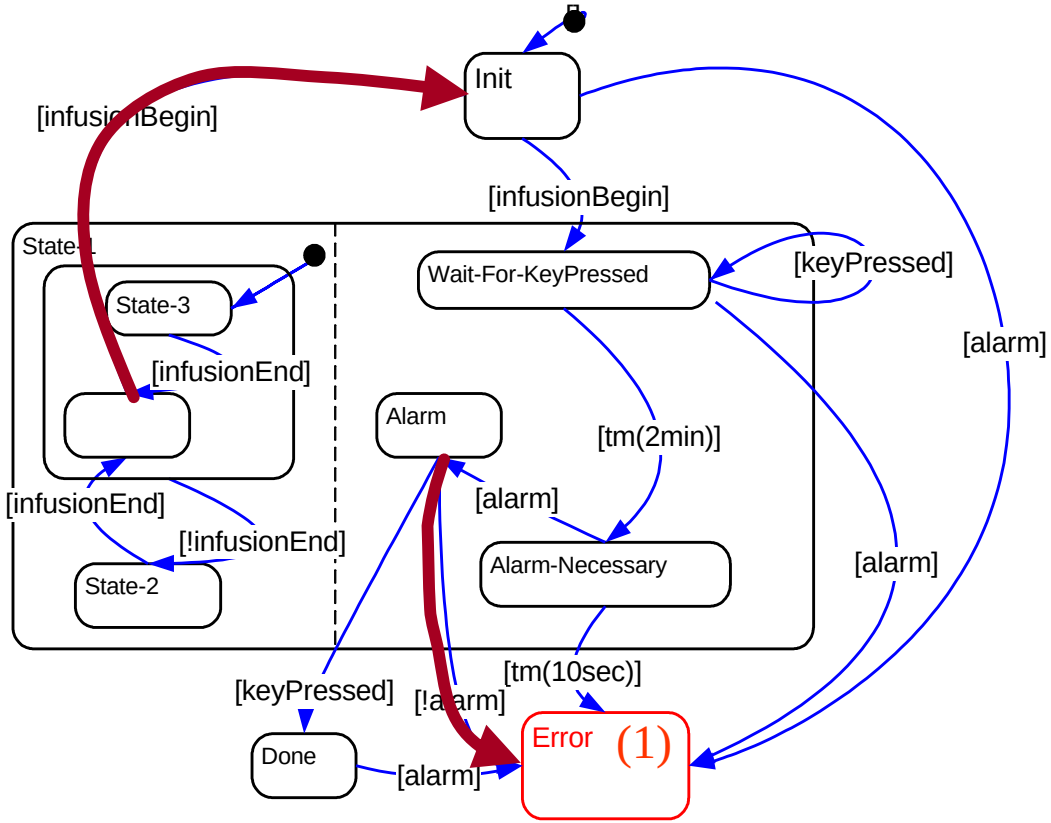
- Defined in terms of an Equivalent NFA (**ENFA**).



TLCharts: Interlingua-based Semantics

First, consider TLCharts without LTL/regex guards:

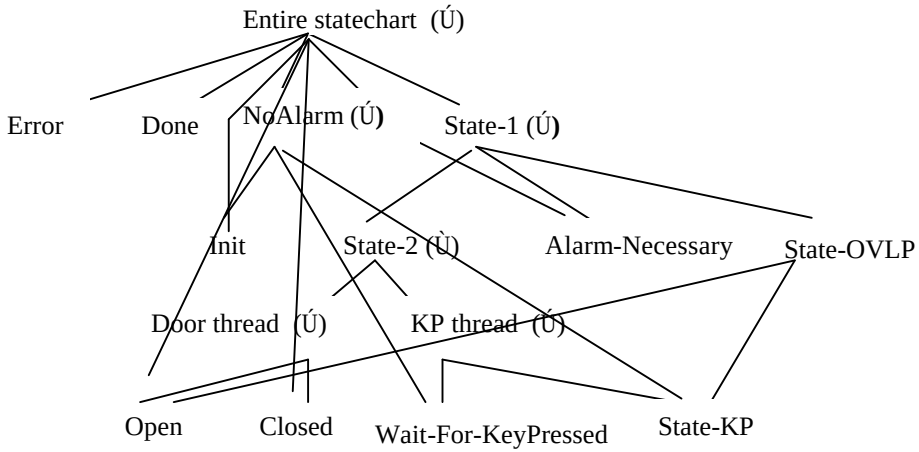
- Extended: has both accepting (good) and rejecting states (Error).
- Accept/reject decision for competing (in case of non-determinism) computations is based on priority scheme.
- This amounts to a NFA with negation.



TLCharts: Interlingua-based Semantics

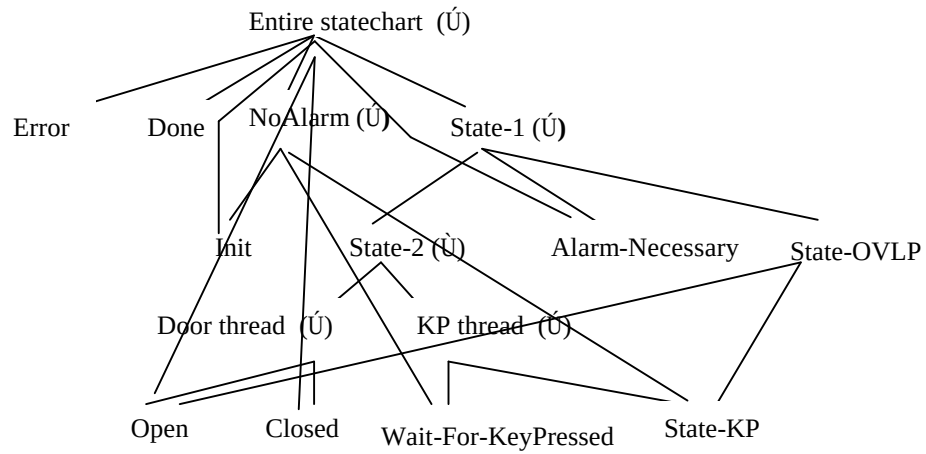
Overlapping states:

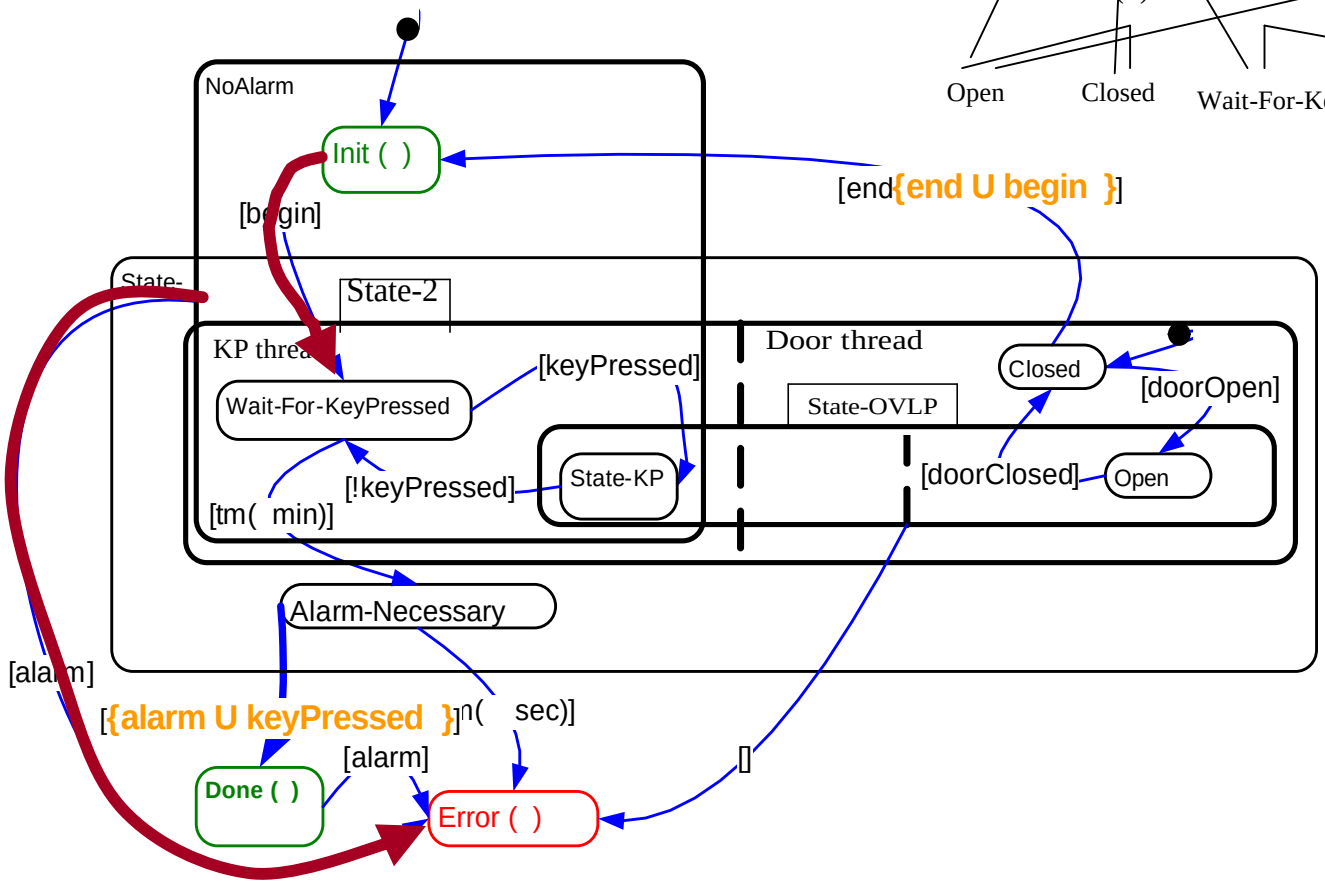
- State DAG instead of state tree.
- State configurations in a computation now contain hierarchical state information.
- Every spanning tree of the state DAG induces a set of competing computations.
- The prioritized accept/reject decision is made over the union of these sets.



TLCharts: Interlingua-based Semantics

Overlapping states:





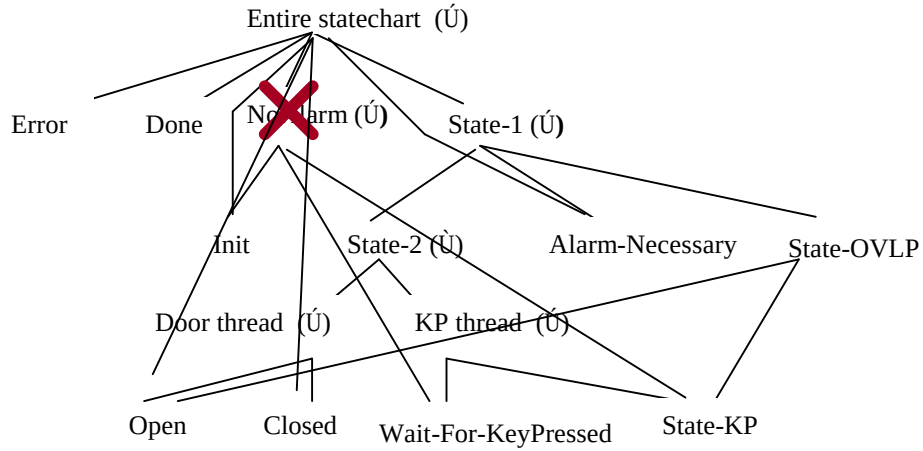
Input:

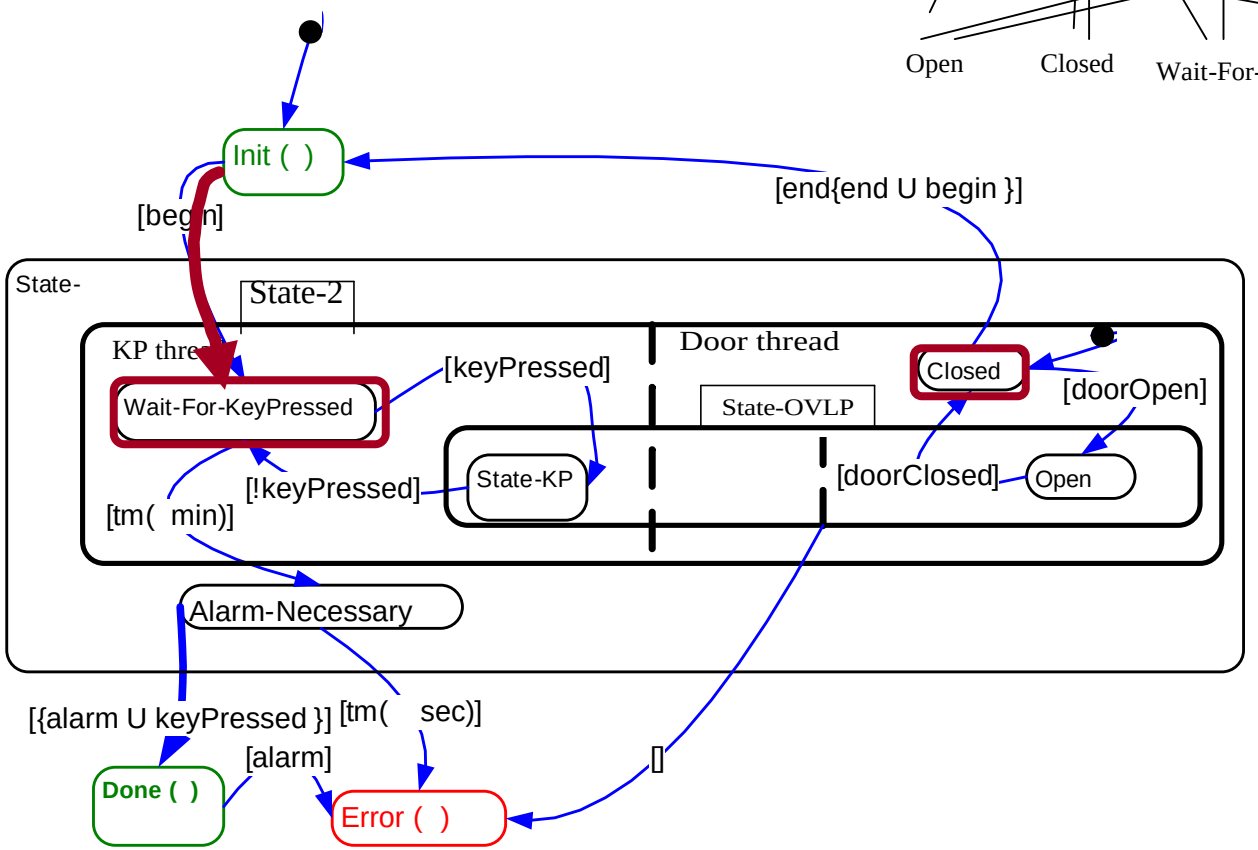
begin.alarm

Doron Drusinsky

TLCharts: Interlingua-based Semantics

Using tree #1:



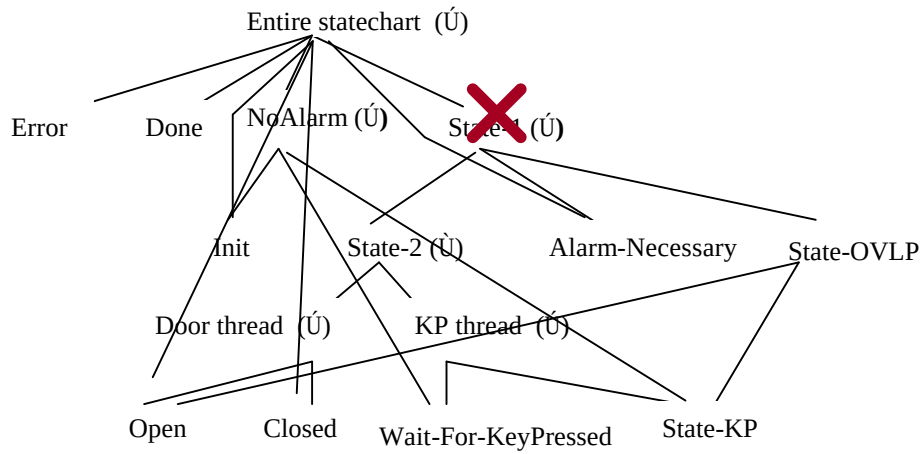


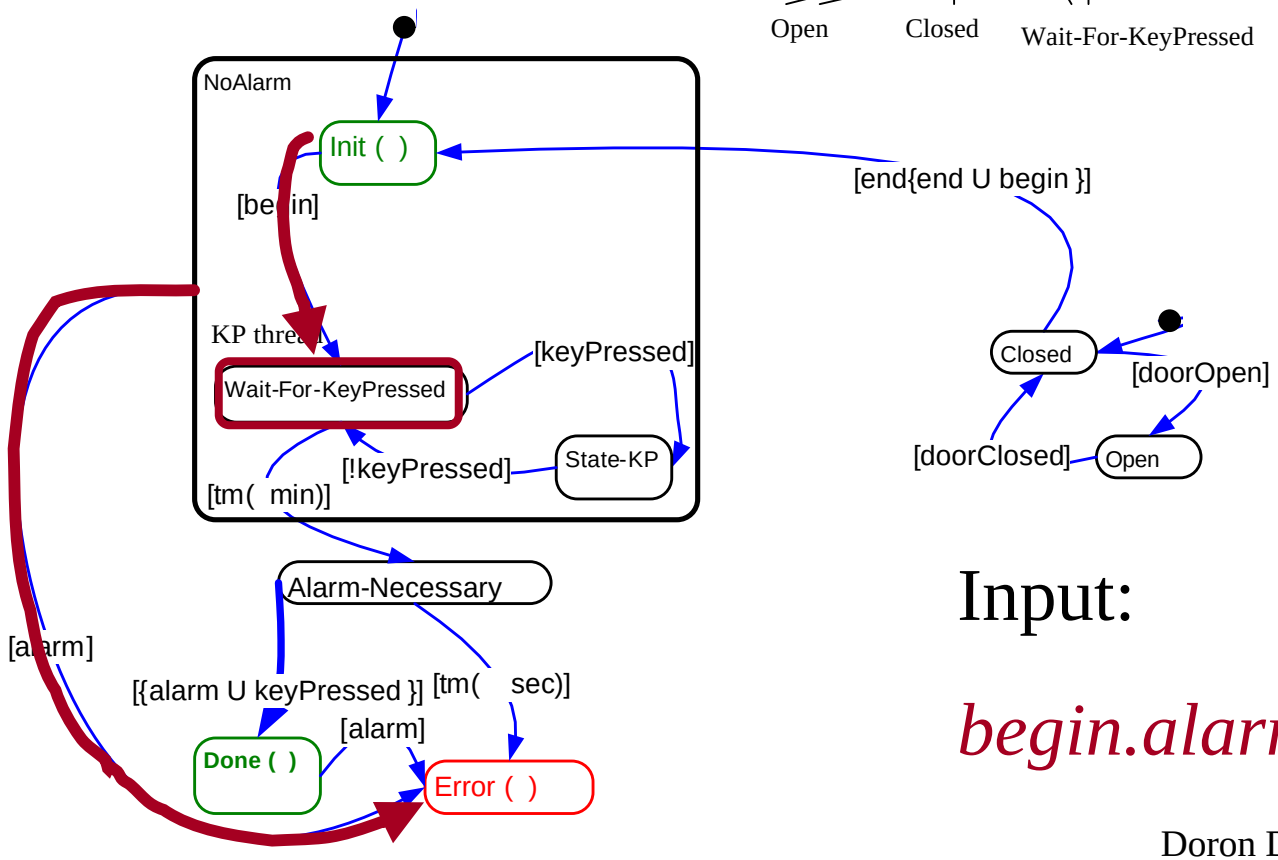
Input:

begin.alarm

TLCharts: Interlingua-based Semantics

Using tree #2:



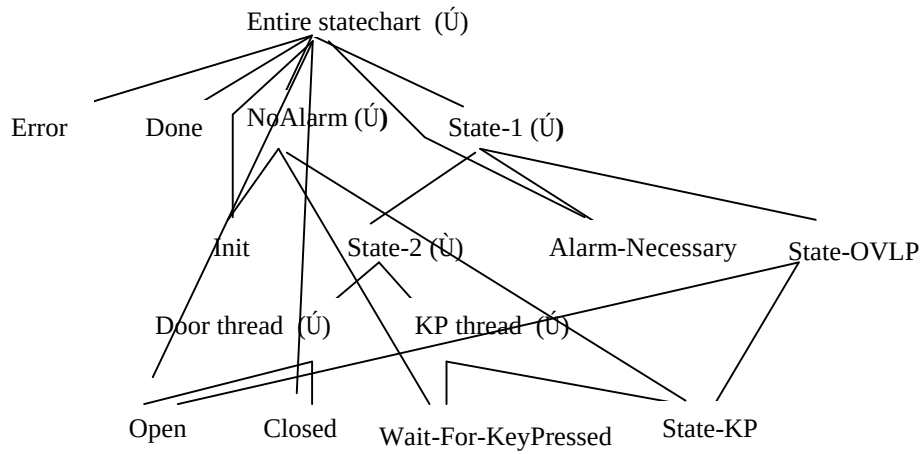


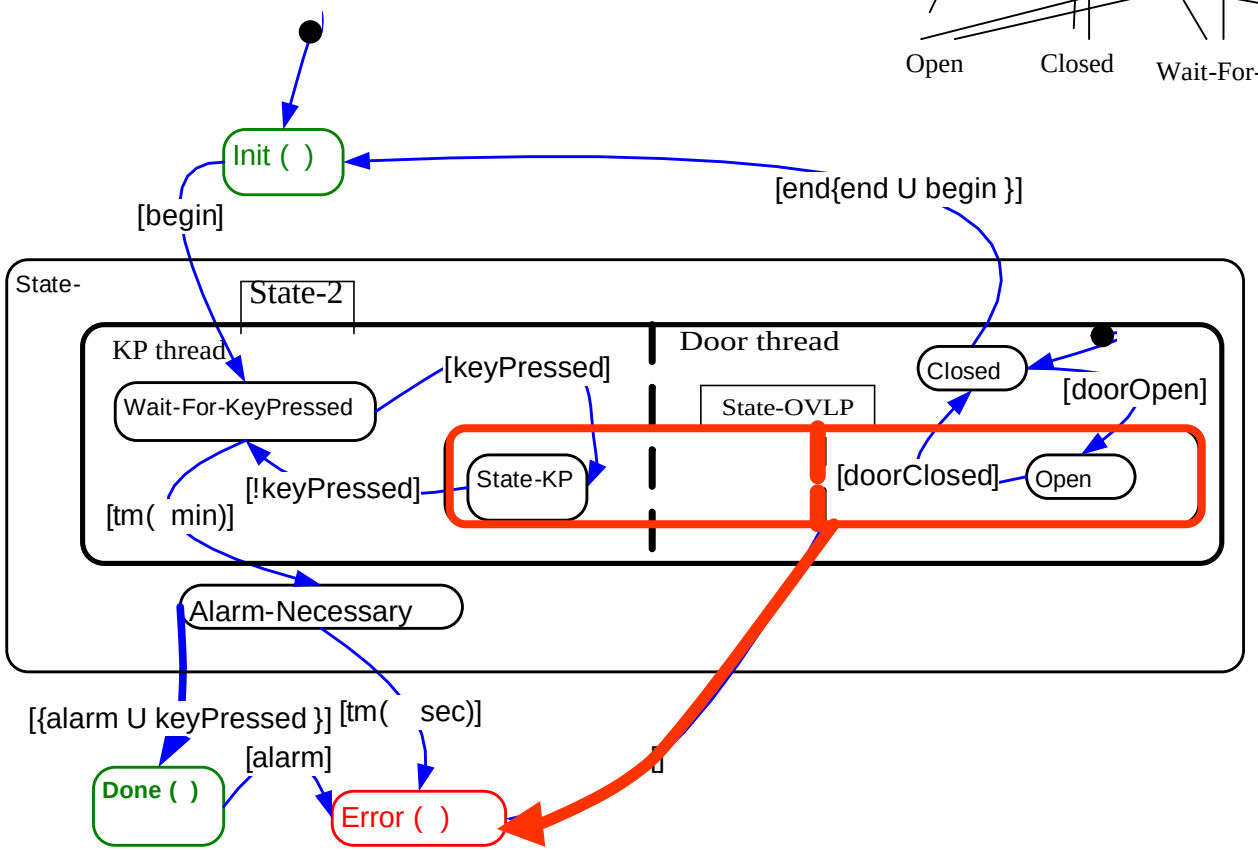
Input:

begin.alarm

TLCharts: Interlingua-based Semantics

Overlapping states:



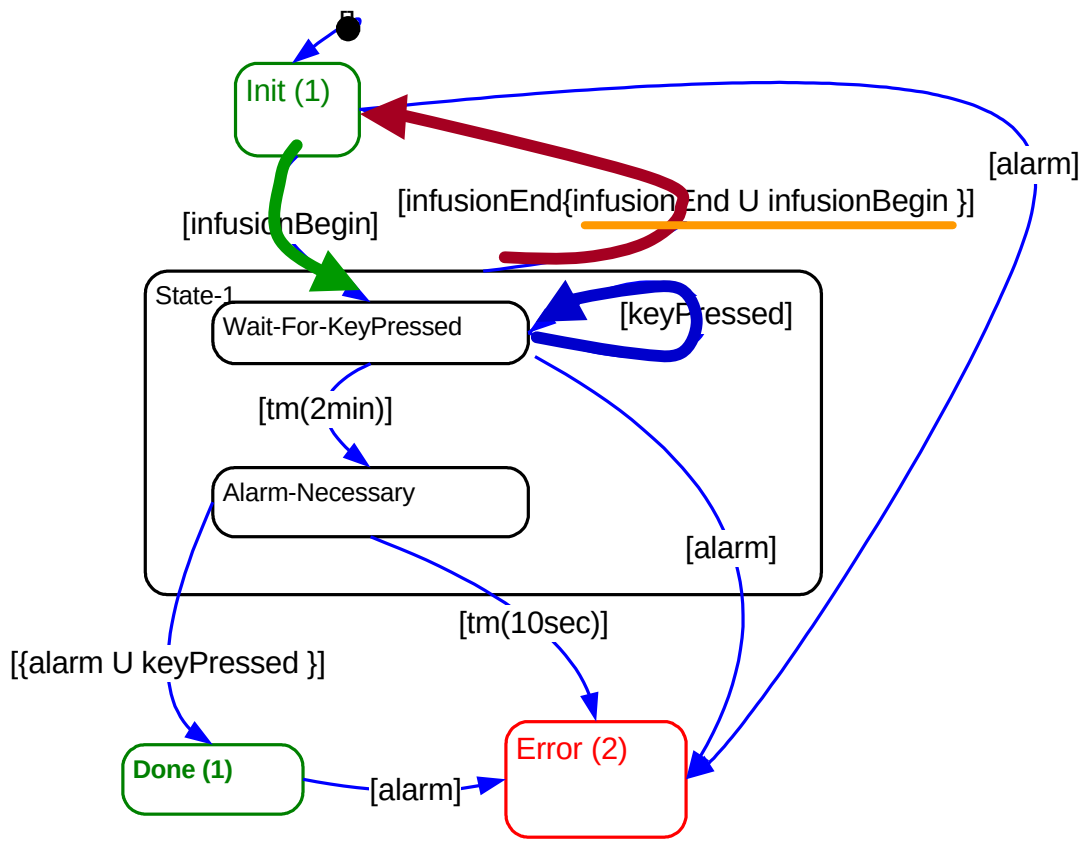


Intuitive meaning:
These two states should never be visited simultaneously

TLCharts: Interlingua-based Semantics

Next, consider LTL/regex guards:

- Defined in terms of an Equivalent NFA (**ENFA**).



infusionBegin .keyPressed .infusionEnd .infusionEnd. infusionEnd. infusionBegin

Transition fires now ↑ **guard**

Thank you